LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-JRNL-573912

Source estimation by full wave form inversion

B. Sjogreen, N. A. Petersson

August 15, 2012

Journal of Scientific Computing

Disclaimer

Source estimation by full wave form inversion

Björn Sjögreen* and N. Anders Petersson*

August 6, 2012

Abstract

Given time-dependent ground motion recordings at a number of receiver stations, we solve the inverse problem for estimating the parameters of the seismic source. The source is modeled as a point moment tensor source, characterized by its location, moment tensor components, the start time, and frequency parameter (rise time) of its source time function. In total, there are 11 unknown parameters. We use a non-linear conjugate gradient algorithm to minimize the full waveform misfit between observed and computed ground motions at the receiver stations.

An important underlying assumption of the minimization problem is that the wave propagation is accurately described by the elastic wave equation in a heterogeneous isotropic material. We use a fourth order accurate finite difference method, developed in [12], to evolve the waves forwards in time. The adjoint wave equation corresponding to the discretized elastic wave equation is used to compute the gradient of the misfit, which is needed by the non-linear conjugated minimization algorithm.

A new source point moment source discretization is derived that guarantees that the Hessian of the misfit is a continuous function of the source location. An efficient approach for calculating the Hessian is also presented. We show how the Hessian can be used to scale the problem to improve the convergence of the non-linear conjugated gradient algorithm. Numerical experiments are presented for estimating the source parameters from synthetic data in a layer over half-space problem (LOH.1), illustrating rapid convergence of the proposed approach.

1 Introduction

This article presents a computational technique for estimating the location, start time, magnitude, and other parameters that describe the source in a seismic event, such as an earthquake, a mine implosion, or an explosion. Our approach is based on using seismographic ground motion recordings together with large scale simulations to estimate the parameters describing the seismic source.

*Center for Applied Scientific Computing, L-422, LLNL, P.O. Box 808, Livermore, CA 94551, USA.
This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

We consider seismic source estimation as a minimization problem constrained by the elastic wave equation subject to appropriate boundary and initial conditions. Our objective of the source estimation is to minimize the difference between the recorded and simulated wave forms. We assume that ground motion observations are recorded at the fixed spatial locations $\mathbf{x}_r$, $r = 1, \ldots, R$, and that three orthogonal components of the displacement are measured as functions of time at all recording stations, denoted by $\mathbf{d}_r(t)$. Let $\mathbf{u}(\mathbf{x}, t)$ be the displacement field governed by the elastic wave equation. The displacement field depends implicitly on the source parameters, which we collect in the Q-dimensional vector $\mathbf{p}$. The continuous minimization problem is defined through the misfit functional

$$\mathcal{X}_c(\mathbf{p}) = \frac{1}{2} \sum_{r=1}^{R} \int_{t=0}^{T} s(t) \left| \mathbf{u}(\mathbf{x}_r, t) - \mathbf{d}_r(t) \right|^2 \, dt, \tag{1}$$

where $s(t) > 0$ is a weight function and $|\mathbf{u}|$ denotes the magnitude of the vector $\mathbf{u}$. Note that the misfit is a non-negative real scalar functional of $\mathbf{u}$, which accounts for differences between the time-dependent wave forms $\mathbf{u}(\mathbf{x}_r, t)$ and $\mathbf{d}_r(t)$ in the time interval $0 \leq t \leq T$. Hence, $\mathcal{X}_c = 0$ implies perfect agreement between the wave forms at all recording stations, i.e., $\mathbf{u}(\mathbf{x}_r, t) = \mathbf{d}_r(t)$ for $0 \leq t \leq T$ and $r = 1, 2, \ldots, R$.

We will minmize the full waveform misfit functional (1) by gradient based optimization, where the gradient is efficiently computed by solving the adjoint problem. This is a technique that has gained popularity for seismic inverse problems in recent years. For a recent example see [16], where both the material inversion and the source inversion problems are addressed. In [16], the the elastic wave equation is solved by a spectral element method. The adjoint problem is formulated for the PDE and is discretized by the spectral element method.

A number of alternative techniques for source estimation exist and are being used for seismic source inversion and material inversion. The different methods can be characterized by three different aspects: how the misfit is defined, how the synthetics are produced (i.e., how $\mathbf{u}$ in (1) is approximated), and the exact algorithm used for the minimization of the misfit.

Traditionally, the inverse problem in seismology is solved by minimizing the misfit between observed and computed arrival times. For an example of this approach, see [13], where both the source inversion and the material inversion problems are addressed. [15] gives a comparison of different misfit functionals, and show some source inversion results. When full waveforms are not used in the misfit, the computation of the solution of the full elastic wave equation is often replaced by ray tracing, or by a geometrical optics (JWKB) approximation. See [1] for a recent example of material inversion based on ray tracing and the so called double-difference traveltime misfit.

The full waveform misfit (1) is difficult to match with good accuracy when the source inversion is made with a material model that is not accurate enough. To compensate for imperfect modeling when working with measured data, it is often necessary to introduce windowing functions in the misfit, that only select a certain part of the measured seismograms, see [16].

For optimization, direct search methods, i.e., minimization algorithms that do not use

the gradient information have also been used to solve the inverse problem in seismology, e.g., the downhill simplex method was used in [11]. See [14] for a review of direct search minimization algorithms. In the same spirit are methods that make use of source-receiver reciprocity to efficiently compute the misfit functional for a large number of different source locations, see [6]. Although these methods are more robust, they tend to require more computational work if the solution is required to high precision.

Optimization methods that make use of both the gradient and the Hessian (or an approximate Hessian), such as Newton and quasi-Newton methods have been successfully used for the inverse problems, see, e.g., [4] for electromagnetic scattering and [10] for resistivity imaging in oil exploration. These methods are made efficient by use of Broyden's update, and might be considered also in seismic source inversion, although, the current study concludes, that with a good preconditioning the non-linear conjugate gradient minimization can be just as efficient as a quasi-Newton method.

We assume that the displacement field $\mathbf{u}(\mathbf{x}, t)$ satisfies the elastic wave equation in the three-dimensional domain Ω, subject to initial and boundary conditions. Here, the boundary is denoted $\Gamma = \Gamma_1 \cup \Gamma_2$. The displacement is governed by

$$\begin{aligned} \rho \mathbf{u}_{tt} &= \nabla \cdot \boldsymbol{\tau}(\mathbf{u}) + \mathbf{f}(\mathbf{x}, t; \mathbf{p}), \quad \mathbf{x} \in \Omega,\ 0 \le t \le T, \\ \mathbf{u}(\mathbf{x}, 0) &= 0, \qquad \mathbf{x} \in \Omega,\ t = 0, \\ \mathbf{u}_t(\mathbf{x}, 0) &= 0, \qquad \mathbf{x} \in \Omega,\ t = 0, \\ \mathbf{n} \cdot \boldsymbol{\tau}(\mathbf{u}) &= 0, \qquad \mathbf{x} \in \Gamma_1,\ 0 \le t \le T, \\ \mathbf{u} &= 0, \qquad \mathbf{x} \in \Gamma_2,\ 0 \le t \le T, \end{aligned} \tag{2}$$

where $\rho(\mathbf{x})$ is the density. We further assume that earth can be described as a heterogeneous isotropic elastic material. The stress tensor $\boldsymbol{\tau}(\mathbf{u})$ is then related to the displacement gradient through

$$\boldsymbol{\tau}(\mathbf{u}) = \lambda \operatorname{div}(\mathbf{u})\, I + \mu\, (\nabla \mathbf{u} + \nabla \mathbf{u}^T), \tag{3}$$

where $\lambda(\mathbf{x})$ and $\mu(\mathbf{x})$ are the first and second Lamé parameters of the material.

The function $\mathbf{f}$ in (2) models the seismic source. For smaller seismic events, it can be modeled by a point moment tensor source,

$$\mathbf{f}(\mathbf{x}, t; \mathbf{p}) = g(t) \mathcal{M} \nabla \delta(\mathbf{x} - \mathbf{x}_*). \tag{4}$$

Here, $\nabla\delta(\mathbf{x})$ is the gradient of the Dirac distribution. The source time function $g(t) = g(t; t_0, \omega_0)$ is assumed to depend on two parameters; a time shift t_0 and a frequency parameter ω_0. The source is located at $\mathbf{x}_* = (x_*,\ y_*,\ z_*)$ and the elements of the symmetric matrix $\mathcal{M}$ are denoted

$$\mathcal{M} = \begin{pmatrix} m^{(xx)} & m^{(xy)} & m^{(xz)} \\ m^{(xy)} & m^{(yy)} & m^{(yz)} \\ m^{(xz)} & m^{(yz)} & m^{(zz)} \end{pmatrix}.$$

Under these assumptions the forcing function $\mathbf{f}$ depends on $Q = 11$ parameters,

$$\mathbf{p} = (x_*,\ y_*,\ z_*,\ m^{(xx)},\ m^{(xy)},\ m^{(xz)},\ m^{(yy)},\ m^{(yz)},\ m^{(zz)},\ t_0,\ \omega_0). \tag{5}$$

The continuous source estimation problem can be stated as the constrained mimization problem

$$\min \mathcal{X}_c(\mathbf{p}), \quad \mathbf{u} \text{ satisfies (2) with forcing } \mathbf{f}(\mathbf{x}, t; \mathbf{p}).$$

Unfortunately, the elastic wave equation can not be solved analytically except in highly idealized situations, such as when the free surface boundary is flat and the material has homogeneous properties. For this reason, we discretize the elastic wave equation by a fourth order accurate finite difference method and solve it numerically. This allows us to account for general heterogeneous material properties and also make our approach extendable to realistic topographies.

The outline of this article is as follows. Section 2 describes the fourth order accurate finite difference discretization, developed in [12], and derives its adjoint discretization. The adjoint property, proved in Sec. 2, is used in Sec. 3 to derive formulas for the gradient and Hessian of the discrete misfit. Section 4 develops a new spatial discretization of the singular source function (4). This discretization is designed to be compatible with fourth order accuracy and be twice continuously differentiable with respect to its postion, $\mathbf{x}_*$. Section 5 describes the initial guess used for the iterative minimization algorithm. The initial guess is obtained by minimizing the arrival time misfit, using ray tracing. In Sec. 6, we perform numerical experiments with the complete source inversion algorithm for a synthetic problem. We investigate how different scaling strategies affect the convergence rate of the minimization algorithm, and demonstrate excellent performance. Finally, conclusions are given in Sec. 7.

2 The discretized problem

2.1 A self-adjoint fourth order accurate finite difference scheme

Consider the elastic wave equation (2) on the rectangular domain $(x, y, z) \in [0, x_{max}] \times [0, y_{max}] \times [0, z_{max}]$, and the time interval $0 \le t \le T$. Let the computational grid be

$$x_i = (i-1)h, \qquad y_j = (j-1)h, \quad \text{and} \quad z_k = (k-1)h,$$

where $h > 0$ is the grid size and i, j, and k are integers in the ranges $i \in [0, N_x + 1]$, $j \in [0, N_y + 1]$, and $k \in [0, N_z + 1]$. The domain sizes are chosen such that $x_{N_x} = x_{max}$, $y_{N_y} = y_{max}$, and $z_{N_z} = z_{max}$. The points with $i = 0$, $i = N_x + 1$, $j = 0$, $j = N_y + 1$, $k = 0$, or $k = N_z + 1$ are ghost points that are used to impose boundary conditions. Time is discretized on the grid $t_n = n\Delta_t$, where $\Delta_t > 0$ is the fixed time step, and $n = 0, 1, \ldots, M$. The time step is chosen such that $t_M = M\Delta_t = T$.

The numerical approximation of the displacement vector $\mathbf{u}(\mathbf{x}, t)$ at grid point (i, j, k) and time level t_n is denoted by $\mathbf{u}^n_{i,j,k} = (u^n_{i,j,k}, v^n_{i,j,k}, w^n_{i,j,k})$. To improve readability, we occasionally supress the subscript or superscript on $\mathbf{u}$, for example by writing $\mathbf{u}^n$ for $\mathbf{u}^n_{i,j,k}$. When convenient we also use the vector index notation $\mathbf{i} = (i, j, k)$ to indicate a spatial grid point index.

In [12], we developed a fourth order accurate symmetric discretization of the divergence of the stress tensor (3). This operator, denoted by $\mathbf{L}_h(\mathbf{u})$, has the property that

$$(\mathbf{v}, \mathbf{L}_h(\mathbf{u}))_h = (\mathbf{L}_h(\mathbf{v}), \mathbf{u})_h, \tag{6}$$

for any two grid functions $\mathbf{u}$ and $\mathbf{v}$ that satisfy the discretized boundary conditions

$$\mathbf{B}(\mathbf{u})_{i,j,k} = 0, \quad \mathbf{x}_{i,j,k} \in \Gamma. \tag{7}$$

The scalar product in (6) is defined by

$$(\mathbf{v}, \mathbf{u})_h = h^3 \sum_{k=1}^{N_z} \sum_{j=1}^{N_y} \sum_{i=1}^{N_x} a_{i,j,k} \langle \mathbf{v}_{i,j,k}, \mathbf{u}_{i,j,k} \rangle,$$

where $a_{i,j,k}$ are positive weights determined from the summation by parts property of $\mathbf{L}_h(\mathbf{u})$ that is needed to enforce (6). Also, $\langle \mathbf{u}, \mathbf{v} \rangle = \sum_{q=1}^{3} u^{(q)} v^{(q)}$, is the inner product between real-valued vectors with three components. Using this notation, the magnitude of $\mathbf{u}$ satisfies $|\mathbf{u}|^2 = \langle \mathbf{u}, \mathbf{u} \rangle$.

We consider boundary operators $\mathbf{B}$ that either discretize free surface or Dirichlet boundary conditions,

$$\mathbf{B}(\mathbf{u}^n)_{i,j,k} = \begin{cases} \mathcal{B}(\mathbf{u}^n)_{i,j,k} \mathbf{n}_{i,j,k}, & \text{Free surface,} \\ \mathbf{u}^n_{i,j,k}, & \text{Dirichlet.} \end{cases}$$

Here, $\mathcal{B}(\mathbf{u})$ is a special difference approximation of the stress tensor on the boundary that matches $\mathbf{L}_h(\mathbf{u})_{i,j,k}$ such that the overall discretization becomes stable. The vector $\mathbf{n}_{i,j,k}$ is the outward boundary normal. A detailed description of the interior and boundary discretizations can be found in [8, 12].

We discretize the elastic wave equation (2) using the fourth order accurate difference method described in [12]. This method computes the displacement field $\mathbf{u}^n$, $n = 1, 2, \ldots, M$, according to Algorithm 1. Note that the grid function $\mathbf{F}(t; \mathbf{p})$ in this algorithm represents a discretization of the singular source term $\mathbf{f}(\mathbf{x}, t; \mathbf{p})$. This discretization will be described in detail in Section 4. The operator $\mathbf{S}_G(\mathbf{u})$ is a damping operator, used in a supergrid sponge layer near artificial non-reflecting boundaries. $\mathbf{S}_G(\mathbf{u})$ is consistent with

$$-\gamma h^4 \Delta_t \left((\sigma^{(x)}(x) \mathbf{u}_{xxt})_{xx} + (\sigma^{(y)}(y) \mathbf{u}_{yyt})_{yy} + (\sigma^{(z)}(z) \mathbf{u}_{zzt})_{zz} \right),$$

where γ is a constant that controls the strength of damping. The scalar, non-negative, taper functions $\sigma^{(x)}, \sigma^{(y)}$, and $\sigma^{(z)}$ are non-zero only in the sponge layers, where they increase from 0 to 1 in the outward direction through the layer. The domain is terminated at the artificial boundary by a homogeneous Dirichlet boundary condition, enforced on the approximation such that the symmetric property

$$(\mathbf{v}, \mathbf{S}_G(\mathbf{u}))_h = (\mathbf{S}_G(\mathbf{v}), \mathbf{u})_h \tag{8}$$

holds, and such that the discrete energy is positive and non-increasing. In the supergrid layer, the material properties ρ, λ, and μ are artificially tapered, such that the compressional and shear wave speeds are smoothly reduced to small values near the boundaries of the layer. The super grid boundary condition, its discretization, and the proof of (8), will be given in [].

Algorithm 1 *4th order accurate predictor-corrector scheme for the elastic wave equation.*

procedure FORWARD($\mathbf{u}, \mathbf{F}$)
  Initial conditions: $\mathbf{u}^0 = \mathbf{0}$ and $\mathbf{u}^{-1} = \mathbf{0}$
  for $n = 0, 1, \ldots, M-1$ **do**
    Predictor step:

$$\mathbf{u}^* = 2\mathbf{u}^n - \mathbf{u}^{n-1} + \frac{\Delta_t^2}{\rho}\big(\mathbf{L}_h(\mathbf{u}^n) + \mathbf{F}(t_n; \mathbf{p})\big)$$

    Impose boundary condition (7) on $\mathbf{u}^*$ to define its ghost point values
    Acceleration: $\mathbf{v}^n = \big(\mathbf{u}^* - 2\mathbf{u}^n + \mathbf{u}^{n-1}\big)/\Delta_t^2$
    Corrector step:

$$\mathbf{u}^{n+1} = \mathbf{u}^* + \frac{\Delta_t^4}{12\rho}\big(\mathbf{L}_h(\mathbf{v}^n) + \mathbf{F}_{tt}(t_n; \mathbf{p})\big) + \mathbf{S}_G(\mathbf{u}^n - \mathbf{u}^{n-1})$$

    Impose boundary condition (7) on $\mathbf{u}^{n+1}$ to define its ghost point values
  end for
end procedure

2.2 The discrete source estimation problem

A straight forward generalization of the continuous formula (1) leads to the discrete misfit functional

$$\mathcal{X}(\mathbf{p}) = \frac{1}{2}\sum_{r=1}^{R}\sum_{n=0}^{M-1} s(t_n)|\mathbf{u}_{\mathbf{i}_r}^n - \mathbf{d}_r(t_n)|^2. \tag{9}$$

As in the continuous problem, $s(t_n) > 0$ is a weight function. We assume that all recording stations coincide with grid points, i.e., $\mathbf{x}_r = \mathbf{x}_{\mathbf{i}_r}$ for some vector index $\mathbf{i}_r = (i_r, j_r, k_r)$. Furthermore, the observed displacements $\mathbf{d}_r(t)$ are assumed to have already been filtered in time such that they only contain motions that can be captured on the computational grid.

Similar to the continuous case, the displacement at the recording stations depends implicitly on the parameter vector $\mathbf{p}$ in the discretized forcing function $\mathbf{F}$. Hence, the discrete source estimation problem can be stated as the constrained minimization problem,

$$\min \mathcal{X}(\mathbf{p}), \quad \mathbf{u}^n \text{ is calculated by Algorithm 1 with forcing } \mathbf{F}(t_n; \mathbf{p}).$$

Given the source parameters $\mathbf{p}$, we can use Algorithm 1 to calculate the solution of the elastic wave equation, which then can be inserted into (9) to evaluate the discrete misfit $\mathcal{X}(\mathbf{p})$.

2.3 The adjoint wave equation

An efficient approach for computing the gradient of the misfit uses the adjoint wave field, $\boldsymbol{\kappa}_{\mathbf{i}}^n$. Let the source term in the adjoint equation be $\mathbf{G}(t_n)$. The adjoint wave field satisfies

the adjoint of the discretized elastic wave equation. A method for calculating $\boldsymbol{\kappa}$ is given in Algorithm 2.

Algorithm 2 *The adjoint scheme for the elastic wave equation.*

procedure ADJOINT($\boldsymbol{\kappa}, \mathbf{G}$)
  Terminal conditions: $\boldsymbol{\kappa}^{M-1} = \mathbf{0}$ and $\boldsymbol{\kappa}^{M} = \mathbf{0}$
  for $n = M-1, M-2, \ldots, 1$ **do**
    Predictor step:

$$\boldsymbol{\kappa}^{*} = 2\boldsymbol{\kappa}^{n} - \boldsymbol{\kappa}^{n+1} + \Delta_t^2 \frac{\mathbf{L}_h(\boldsymbol{\kappa}^{n})}{\rho} \tag{10}$$

    Impose boundary condition (7) on $\boldsymbol{\kappa}^{*}$ to define its ghost point values
    Compute acceleration: $\boldsymbol{\zeta}^{n} = \big(\boldsymbol{\kappa}^{*} - 2\boldsymbol{\kappa}^{n} + \boldsymbol{\kappa}^{n+1}\big)/\Delta_t^2$
    Corrector step:

$$\boldsymbol{\kappa}^{n-1} = \boldsymbol{\kappa}^{*} + \frac{\Delta_t^4}{12}\frac{\mathbf{L}_h(\boldsymbol{\zeta}^{n})}{\rho} + \frac{\Delta_t^2}{\rho}\mathbf{G}(t_n) - \mathbf{S}_G(\boldsymbol{\kappa}^{n+1} - \boldsymbol{\kappa}^{n}), \tag{11}$$

    Impose boundary condition (7) on $\boldsymbol{\kappa}^{n-1}$ to define its ghost point values
  end for
end procedure

The adjoint property is made precise in the following theorem.

Theorem 1. *Let* $\mathbf{F}$ *(with second time derivative* $\mathbf{F}_{tt}$*) be the source term in the discretized elastic wave equation, and use Algorithm 1 to calculate* $\mathbf{u}$*. Furthermore, let* $\mathbf{G}$ *be the source term in the adjoint wave equation and use Algorithm 2 to calculate* $\boldsymbol{\kappa}$ *(with acceleration* $\boldsymbol{\zeta}$*). Then the grid functions* $\mathbf{u}$ *and* $\boldsymbol{\kappa}$ *are adjoint in the sense that*

$$\sum_{n=0}^{M-1} (\mathbf{G}^{n}, \mathbf{u}^{n})_h = \sum_{n=0}^{M-1} \left(\boldsymbol{\kappa}^{n}, \mathbf{F}^{n} + \frac{\Delta_t^2}{12}\mathbf{F}_{tt}^{n} \right)_h + \frac{\Delta_t^2}{12} \sum_{n=0}^{M-1} (\boldsymbol{\zeta}^{n}, \mathbf{F}^{n})_h \,. \tag{12}$$

Proof: See Appendix A.

3 Minimizing the misfit

We use a preconditioned Fletcher-Reeves method to minimize the discrete misfit. This technique generalizes the conjugate gradient method to non-quadratic problems, see, e.g., [7]. The preconditioning corresponds to a change of variables, $\hat{\mathbf{p}} = S\mathbf{p}$, where S is a non-singular matrix. The change of variables is introduced to improve the convergence properties of the Fletcher-Reeves algorithm. We first formulate the minimization algorithm in the scaled variables, and then transform it back to the original variables. The resulting algorithm with m restarts, and where the parameter vector $\mathbf{p}$ has Q components is given in Algorithm 3. The algorithm terminates after all restarts have been completed, or when the maximim norm of the scaled gradient is smaller than the tolerance $\theta \ll 1$. In

practice we usually set $\theta = 10^{-12}$. Note that the algorithm is given for a general preconditioning matrix S. When S is diagonal, $S^T S = S^2$. The Fletcher-Reeves algorithm uses

Algorithm 3 *The preconditioned Fletcher-Reeves method. Here,* $\nabla \mathcal{X}_k = \nabla \mathcal{X}(\mathbf{p}_k)$.

procedure PRECOND-FLETCHER-REEVES($\mathbf{p}_0$)
  for $r = 1, 2, \ldots, m$ **do**
    Initial search direction: $\mathbf{q}_0 = -(S^T S)^{-1} \nabla \mathcal{X}(\mathbf{p}_0)$
    for $k = 0, 1, \ldots, Q-1$ **do**
      Line search: find steplength α_k that minimizes $\mathcal{X}(\mathbf{p}_k + \alpha_k \mathbf{q}_k)$
      Next parameter vector: $\mathbf{p}_{k+1} = \mathbf{p}_k + \alpha_k \mathbf{q}_k$
      Compute β_k:

$$\beta_k = \frac{\nabla \mathcal{X}_{k+1}^T (S^T S)^{-1} \nabla \mathcal{X}_{k+1}}{\nabla \mathcal{X}_k^T (S^T S)^{-1} \nabla \mathcal{X}_k}$$

      Next search direction: $\mathbf{q}_{k+1} = -(S^T S)^{-1} \nabla \mathcal{X}(\mathbf{p}_{k+1}) + \beta_k \mathbf{q}_k$
      if $||S^{-1} \nabla \mathcal{X}_{k+1}||_\infty < \theta$ **then**
        $\mathbf{p}_0 = \mathbf{p}_{k+1}$
        return
      end if
    end for
    Initial guess for next outer iteration $\mathbf{p}_0 = \mathbf{p}_Q$
  end for
end procedure

the gradient of the misfit with respect to the components of the parameter vector $\mathbf{p}$. It is defined by

$$\nabla \mathcal{X}(\mathbf{p}) = \left(\frac{\partial \mathcal{X}}{\partial p_1}, \frac{\partial \mathcal{X}}{\partial p_2}, \ldots, \frac{\partial \mathcal{X}}{\partial p_Q} \right)^T.$$

In section 3.1, we will discuss an efficient approach for calculating all components of the gradient by solving one adjoint wave equation.

The convergence properties of the Fletcher-Reeves algorithm depend on the properties of the scaled Hessian matrix, with elements $\hat{H}_{i,j} = \partial^2 \mathcal{X} / \partial \hat{p}_i \partial \hat{p}_j$. In matrix notation, we have

$$\hat{H} = S^{-T} H S^{-1}, \quad H_{i,j} = \frac{\partial^2 \mathcal{X}}{\partial p_i \partial p_j},$$

where H is the unscaled Hessian. Superscript T denotes transpose. The minimization algorithms is not guaranteed to converge unless the Hessian matrix is well defined, i.e., the misfit function must be twice continuously differentiable with respect to $\mathbf{p}$. It is straightforward to see that the displacement field depends linearly on the matrix elements of $\mathcal{M}$. Hence $\mathbf{u}$ and thereby $\mathcal{X}$ are infinitely differentiable with respect to the elements of $\mathcal{M}$. We will assume that the time function depends on t_0 through a time shift $g(t; t_0, \omega_0) = \tilde{g}(t - t_0; \omega_0)$. Because the source term $\mathbf{F}$ enters into the finite difference scheme with two time derivatives (see Algorithm 1), a requirement for the Hessian to be defined is that

$\tilde{g}(t;\omega_0)$ is four times differentible with respect to t and twice differentiable with respect to ω_0. In this article, $\tilde{g}$ is assumed to have this regularity. The spatial discretization and regularity of the moment tensor source with respect to its location, $\mathbf{x}_*$ will be described in section 4.

A crucial component of the preconditioned Fletcher-Reeves algorithm is the line search algorithm, i.e., to minimize $\mathcal{X}(\mathbf{p}_k + \alpha\mathbf{q}_k)$ with respect to the step length α. In our implementation, we use a backtracking algorithm called A6.3.1 in [3], with the minor modification that the full step is $\mathbf{p}_k + \alpha_s\mathbf{q}_k$, instead of $\mathbf{p}_k + \mathbf{q}_k$, as assumed in [3]. The full step size α_s is taken from the linear conjugated gradient algorithm, which assumes that $\mathcal{X}(\mathbf{p})$ is a quadratic function of $\mathbf{p}$,

$$\alpha_s = -\frac{\nabla\mathcal{X}_k^T\mathbf{q}_k}{\mathbf{q}_k^T H_k\mathbf{q}_k}, \tag{13}$$

In our case, the Hessian is evaluated at $\mathbf{p}_k$, i.e., $H_k = H(\mathbf{p}_k)$. In section 3.2, we present an algorithm for evaluating $\mathbf{q}_k^T H_k\mathbf{q}_k$ that only requires one additional wave equation to be solved. In most iterations of the Fletcher-Reeves algorithm, the full step length α_s yields an acceptable approximation of the minima in the line search. Hence, the backtracking is rarely invoked. The "typical x" vector, needed in backtracking algorithm A6.3.1 (see [3]), is taken as the inverse of the diagonal elements of the scaling matrix S.

3.1 The gradient of the misfit

Straightforward differentiation of (9) gives

$$\frac{\partial\mathcal{X}}{\partial p_j} = \sum_{r=1}^{R}\sum_{n=0}^{M-1} s(t_n)\left\langle \mathbf{u}_{\mathbf{i}_r}^n - \mathbf{d}_r(t_n), \frac{\partial\mathbf{u}_{\mathbf{i}_r}^n}{\partial p_j}\right\rangle. \tag{14}$$

Note that the material properties ρ, μ, and λ do not depend on p_j. By differentiating the difference scheme for $\mathbf{u}$ with respect to p_j, we see that $\partial\mathbf{u}/\partial p_j$ could be calculated with the same finite difference scheme as used for computing $\mathbf{u}$, if the source term $\mathbf{F}$ is replaced by $\partial\mathbf{F}/\partial p_j$. However, to compute the gradient of $\mathcal{X}$ with this technique, it would be necessary to solve the elastic wave equation with 11 different forcing functions, where each forcing corresponds to one component of $\partial\mathcal{X}/\partial p_j$.

A more efficient way of computing the gradient of the misfit is based on solving the adjoint wave equation. In this approach, we define the adjoint source in (11) as

$$\mathbf{G}_{\mathbf{i}}^n = \sum_{r=1}^{R} s(t_n)\left(\mathbf{u}_{\mathbf{i}_r}^n - \mathbf{d}_r(t_n)\right)\delta_{\mathbf{i},\mathbf{i}_r}/(h^3 a_{\mathbf{i}}), \tag{15}$$

where

$$\delta_{\mathbf{i},\mathbf{j}} = \begin{cases} 1, & \mathbf{i} = \mathbf{j}, \\ 0, & \text{otherwise.} \end{cases}$$

Inserting (15) into (14) shows that the gradient of the misfit can be written

$$\frac{\partial\mathcal{X}}{\partial p_j} = \sum_{n=0}^{M-1}\left(\mathbf{G}^n, \frac{\partial\mathbf{u}^n}{\partial p_j}\right)_h.$$

Because $\partial \mathbf{u}/\partial p_j$ satisfies the forward finite difference scheme with source term $\partial \mathbf{F}/\partial p_j$, we can apply Theorem 1 to obtain

$$\frac{\partial \mathcal{X}}{\partial p_j} = \sum_{n=0}^{M-1} \left(\boldsymbol{\kappa}^n, \frac{\partial \mathbf{F}^n}{\partial p_j} + \frac{\Delta_t^2}{12} \frac{\partial \mathbf{F}_{tt}^n}{\partial p_j} \right)_h + \frac{\Delta_t^2}{12} \sum_{n=0}^{M-1} \left(\boldsymbol{\zeta}^n, \frac{\partial \mathbf{F}^n}{\partial p_j} \right)_h . \tag{16}$$

Equation (16) allows us to compute all components of the misfit gradient from the adjoint wave field $\boldsymbol{\kappa}_{\mathbf{i}}^n$. The scalar products with the gradients of $\mathbf{F}$ can be assembled during the adjoint solve, which imposes a negligible computational cost compared with the cost of solving the discrete elastic wave equation.

3.2 Calculating the Hessian and $\mathbf{q}^T H \mathbf{q}$

The Hessian matrix plays an important role in gradient based optimization. For example, the condition number of the Hessian governs the convergence rate of the conjugate gradient algorithm, and the Hessian can be used to construct a preconditioner.

To compute the Hessian, we differentiate (14) with respect to p_k to obtain

$$\begin{aligned} H_{k,j} &:= \frac{\partial}{\partial p_k} \left(\frac{\partial \mathcal{X}}{\partial p_j} \right) = \sum_{r=1}^{R} \sum_{n=0}^{M-1} s(t_n) \frac{\partial}{\partial p_k} \left\langle \mathbf{u}_{\mathbf{i}_r}^n - \mathbf{d}_r(t_n), \frac{\partial \mathbf{u}_{\mathbf{i}_r}^n}{\partial p_j} \right\rangle \\ &= \sum_{r=1}^{R} \sum_{n=0}^{M-1} s(t_n) \left\langle \frac{\partial \mathbf{u}_{\mathbf{i}_r}^n}{\partial p_j}, \frac{\partial \mathbf{u}_{\mathbf{i}_r}^n}{\partial p_k} \right\rangle - \sum_{r=1}^{R} \sum_{n=0}^{M-1} s(t_n) \left\langle \mathbf{u}_{\mathbf{i}_r}^n - \mathbf{d}_r(t_n), \frac{\partial^2 \mathbf{u}_{\mathbf{i}_r}^n}{\partial p_j \partial p_k} \right\rangle . \end{aligned} \tag{17}$$

We decompose the Hessian into two parts, $H = H^{(1)} - H^{(2)}$, where

$$H_{j,k}^{(1)} := \sum_{r=1}^{R} \sum_{n=0}^{M-1} s(t_n) \left\langle \frac{\partial \mathbf{u}_{\mathbf{i}_r}^n}{\partial p_j}, \frac{\partial \mathbf{u}_{\mathbf{i}_r}^n}{\partial p_k} \right\rangle , \tag{18}$$

$$H_{j,k}^{(2)} := \sum_{r=1}^{R} \sum_{n=0}^{M-1} s(t_n) \left\langle \mathbf{u}_{\mathbf{i}_r}^n - \mathbf{d}_r(t_n), \frac{\partial^2 \mathbf{u}_{\mathbf{i}_r}^n}{\partial p_j \partial p_k} \right\rangle . \tag{19}$$

By noting the similarities between (14) and (19), we see that the matrix $H^{(2)}$ can also be computed using the adjoint $\boldsymbol{\kappa}$. We arrive at the formula

$$H_{j,k}^{(2)} = \sum_{n=0}^{M-1} \left(\boldsymbol{\kappa}^n, \frac{\partial^2 \mathbf{F}^n}{\partial p_j \partial p_k} + \frac{\Delta_t^2}{12} \frac{\partial^2 \mathbf{F}_{tt}^n}{\partial p_j \partial p_k} \right)_h + \frac{\Delta_t^2}{12} \sum_{n=0}^{M-1} \left(\boldsymbol{\zeta}^n, \frac{\partial^2 \mathbf{F}^n}{\partial p_j \partial p_k} \right)_h , \tag{20}$$

corresponding to (16), but where the first derivative of the source term is replaced by its second derivative. Hence, we can obtain $H^{(2)}$ by accumulating addititional scalar products with $\boldsymbol{\kappa}^n$. Therefore, the computation of $H^{(2)}$ does not require any additional solutions of the elastic wave equation. However, to compute $H^{(1)}$ it is necessary to solve 11 elastic wave equations, to obtain the individual terms $\partial \mathbf{u}_{\mathbf{i}}^n / \partial p_j$ corresponding to the source terms $\partial \mathbf{F}/\partial p_j$. These grid functions are then used to form $H^{(1)}$ directly, according to (18).

The higher computational cost of the Hessian makes it prohibitively expensive to evaluate at each iteration in Algorithm 3. However, as we will see below, it is highly advantageous to compute the Hessian at least once, and use it as a preconditioner throughout the iterations.

The step length calculation (13) for α_s requires the computation of the scalar quantity $\mathbf{q}^T H \mathbf{q}$, where $\mathbf{q}$ is a vector with $Q = 11$ components. As before, we decompose the Hessian into $H = H^{(1)} - H^{(2)}$. The second term, $\mathbf{q}^T H^{(2)} \mathbf{q}$, is directly available by evaluating $H^{(2)}$ as described above. For the first term, we note that

$$\mathbf{q}^T H^{(1)} \mathbf{q} = \sum_{j=1,k=1}^{Q} \sum_{r=1}^{R} \sum_{n=0}^{M-1} s(t_n) \left\langle q_j \frac{\partial \mathbf{u}_{\mathbf{i}_r}^n}{\partial p_j}, \frac{\partial \mathbf{u}_{\mathbf{i}_r}^n}{\partial p_k} q_k \right\rangle .$$

Let $\tilde{\mathbf{u}}_{\mathbf{i}}^n$ denote the solution obtained by solving the discretized elastic wave equation with the source term $\sum_j q_j \frac{\partial \mathbf{F}(t_n)}{\partial p_j}$. It then holds that $\tilde{\mathbf{u}}_{\mathbf{i}}^n = \sum_j q_j \frac{\partial \mathbf{u}_{\mathbf{i}}^n}{\partial p_j}$, and hence,

$$\mathbf{q}^T H^{(1)} \mathbf{q} = \sum_{r=1}^{R} \sum_{n=0}^{M-1} s(t_n) \left\langle \tilde{\mathbf{u}}_{\mathbf{i}_r}^n, \tilde{\mathbf{u}}_{\mathbf{i}_r}^n \right\rangle ,$$

can be assembled during the time stepping for calculating $\tilde{\mathbf{u}}^n$. Hence, the additional cost for calculating $\mathbf{q}^T H \mathbf{q}$ amounts to solving one elastic wave equation. This is the same computational cost as for estimating the step length by an approximate difference quotient, such as that used in [5].

4 Discretizing the singular source term

The gradient of the Dirac distribution in the seismic source term (4) can discretized based on the discretization of a one-dimensional Dirac distributions $\delta(x - x_*)$ and its derivative $\delta'(x - x_*)$. For all smooth, compactly supported functions of one variable $\varphi(x)$, we have

$$\int \varphi(x) \delta(x - x_*)\, dx = \varphi(x_*) \qquad \int \varphi(x) \frac{d\delta}{dx}(x - x_*)\, dx = -\frac{d\varphi}{dx}(x_*). \tag{21}$$

Our approach is based on the technique in [9], which approximates the singular sources numerically by grid functions that satisfy (21) in a discrete scalar product for all polynomial functions up to order $q > 0$, leading to $q + 1$ moment conditions. The required order is related to the order of accuracy in the approximation of the differential equation. Because we use gradient based optimization to solve for the source location, the source term discretization must be twice continuously differentiable with respect to $\mathbf{x}_*$. As a result, the moment conditions must be augmented by additional continuity conditions.

For a fourth order scheme, the moment conditions for δ should be satisfied for the functions $\varphi(x) = x^k$, $k = 0, \ldots, 3$, and the moment conditions for δ' should be satisfied for $k = 0, \ldots, 4$. Details are given in [9]. To make the technique easier to implement, we use discretizations that satisfy the moment conditions for $k = 0, \ldots, 4$, both for δ and δ'.

We describe the discretization of δ and its derivative in one space dimension. The multi-dimensional approximation can be obtained in a straightforward way by Cartesian products of the one dimensional discretizations. Let the one-dimensional grid be $x_j = jh$, $j = 0, \ldots, N+1$, and define the scalar product by $(u, v)_{h1} = h \sum_{j=1}^{N} u_j v_j$. Furthermore, let $\tilde{b}(x_*, j_s)_j$ denote a preliminary approximation of $\delta(x - x_*)$, which is centered at grid point j_s. The five moment conditions

$$\left(x^k, \tilde{b}\right)_{h1} = (x_*)^k, \quad k = 0, \ldots, 4, \tag{22}$$

can be satisfied by using five non-zero coefficients in $\tilde{b}(x_*, j_s)$. We choose a compact stencil that is zero for $j < j_s - 2$ and $j > j_s + 2$. The same grid points are used for source locations at $|x_* - x_{j_s}| < h/2$. Within this interval, the source discretization is infinitely differentiable with respect to the source location, since the coefficients depend on the source location through the right hand side of (22), which consists of polynomials in x_*. However, if $x_* = x_{j_s} + h/2 + \varepsilon$, the stencil will be centered around grid point x_{j_s} for $\varepsilon < 0$, but around grid point x_{j_s+1} for $\varepsilon \geq 0$. Unfortunately, the coefficients are not continuously differentiable at $\epsilon = 0$.

To obtain a source discretization that has two continuous derivatives with respect to the source location, we use a smoothly weighted average of the five point stencils centered at j_s and $j_s + 1$, respectively. We locate the averaged source discretization on the grid by choosing the index j_s such that

$$x_{j_s} \leq x_* < x_{j_s+1}, \quad \text{i.e.,} \quad x_* = x_{j_s} + \nu h, \quad 0 \leq \nu < 1.$$

We have $\nu = (x_* - x_{j_s})/h$ and define the blending function

$$\psi(\nu) = \begin{cases} 0, & \nu < 0, \\ 10\nu^3 - 15\nu^4 + 6\nu^5, & 0 \leq \nu < 1, \\ 1, & \nu \geq 1. \end{cases}$$

The function $\psi(\nu)$ is monotonically increasing for $0 \leq \nu < 1$ and has two continuous derivatives at the break points $\nu = 0$ and $\nu = 1$. The averaged source discretization is then defined as the six point stencil

$$b(\nu, j_s) = (1 - \psi(\nu))\, \tilde{b}(x_*, j_s) + \psi(\nu) \tilde{b}(x_*, j_s + 1), \quad 0 \leq \nu < 1, \tag{23}$$

which obviously is continuously differentiable in $0 \leq \nu < 1$. Note that the coefficient in front of $\tilde{b}(x_*, j_s)$ goes to zero as $\nu \to 1$, and the coefficient in front of $\tilde{b}(x_*, j_s + 1)$ vanishes as $\nu \to 0$.

Away from the source, $b(\nu, j_s)_j = 0$ for $j < j_s - 2$ or $j > j_s + 3$. To verify that the source discretization is a continuous function of the source location, consider the limiting case $\nu \to 1$. Continuity means that the grid function centered at j_s with $\nu = 1$ is identical to the grid function centered at $j_s + 1$ with $\nu = 0$, i.e.,

$$b(1, j_s)_j = b(0, j_s + 1)_j.$$

Since $b(0,j_s)_j$ only depends on j_s through the index j, we have $b(0,j_s+1)_j = b(0,j_s)_{j-1}$. The same argument holds for the continuity of any derivative of b with respect to the source location. Hence, the grid function b and its first two derivatives are continuous functions of x_* if

$$\frac{\partial^k b}{\partial \nu^k}(1,j_s)_j = \frac{\partial^k b}{\partial \nu^k}(0,j_s)_{j-1}, \qquad k = 0,1,2. \tag{24}$$

Note that the grid function b satisfies the moment conditions (22) because both $\tilde{b}(x_*,j_s)$ and $\tilde{b}(x_*,j_s+1)$ satisfy those conditions. After some algebra, we find that the coefficients in the six point stencil (23) are given by

$$b(\nu,j_s)_{j_s-2} = \frac{1}{h}\left(\frac{1}{12}\nu - \frac{1}{24}\nu^2 - \frac{1}{12}\nu^3 - \frac{19}{24}\nu^4 + P(\nu)\right), \tag{25}$$

$$b(\nu,j_s)_{j_s-1} = \frac{1}{h}\left(-\frac{2}{3}\nu + \frac{2}{3}\nu^2 + \frac{1}{6}\nu^3 + 4\nu^4 - 5P(\nu)\right), \tag{26}$$

$$b(\nu,j_s)_{j_s} = \frac{1}{h}\left(1 - \frac{5}{4}\nu^2 - \frac{97}{12}\nu^4 + 10P(\nu)\right), \tag{27}$$

$$b(\nu,j_s)_{j_s+1} = \frac{1}{h}\left(\frac{2}{3}\nu + \frac{2}{3}\nu^2 - \frac{1}{6}\nu^3 + \frac{49}{6}\nu^4 - 10P(\nu)\right), \tag{28}$$

$$b(\nu,j_s)_{j_s+2} = \frac{1}{h}\left(-\frac{1}{12}\nu - \frac{1}{24}\nu^2 + \frac{1}{12}\nu^3 - \frac{33}{8}\nu^4 + 5P(\nu)\right), \tag{29}$$

$$b(\nu,j_s)_{j_s+3} = \frac{1}{h}\left(\frac{5}{6}\nu^4 - P(\nu)\right), \tag{30}$$

where

$$P(\nu) = \frac{5}{3}\nu^5 - \frac{7}{24}\nu^6 - \frac{17}{12}\nu^7 + \frac{9}{8}\nu^8 - \frac{1}{4}\nu^9.$$

Symbolic formula manipulation software (or another tedious calculation) can be used to verify that the continuity conditions (24) are satisfied by the above grid function.

Let $g(j_s,\nu)_j$ denote the grid function approximating the derivative of the Dirac distribution, $\delta'(x-x_*)$. Following the same approach as above, we arrive at the six point stencil

$$e(\nu,j_s)_{j_s-2} = \frac{1}{h^2}\left(-\frac{1}{12} + \frac{1}{12}\nu + \frac{1}{4}\nu^2 + \frac{2}{3}\nu^3 + R(\nu)\right), \tag{31}$$

$$e(\nu,j_s)_{j_s-1} = \frac{1}{h^2}\left(\frac{2}{3} - \frac{4}{3}\nu - \frac{1}{2}\nu^2 - \frac{7}{2}\nu^3 - 5R(\nu)\right), \tag{32}$$

$$e(\nu,j_s)_{j_s} = \frac{1}{h^2}\left(\frac{5}{2}\nu + \frac{22}{3}\nu^3 + 10R(\nu)\right), \tag{33}$$

$$e(\nu,j_s)_{j_s+1} = \frac{1}{h^2}\left(-\frac{2}{3} - \frac{4}{3}\nu + \frac{1}{2}\nu^2 - \frac{23}{3}\nu^3 - 10R(\nu)\right), \tag{34}$$

$$e(\nu,j_s)_{j_s+2} = \frac{1}{h^2}\left(\frac{1}{12} + \frac{1}{12}\nu - \frac{1}{4}\nu^2 + 4\nu^3 + 5R(\nu)\right), \tag{35}$$

$$e(\nu,j_s)_{j_s+3} = \frac{1}{h^2}\left(-\frac{5}{6}\nu^3 - R(\nu)\right). \tag{36}$$

Here, the polynomial R is given by

$$R(\nu) = -\frac{25}{12}\nu^4 - \frac{3}{4}\nu^5 + \frac{59}{12}\nu^6 - 4\nu^7 + \nu^8,$$

and $e(\nu, j_s)_j = 0$ for $j < j_s - 2$ or $j > j_s + 3$.

It can be verified that the grid function e satisfies the moment conditions for a fourth order accurate discretization of $\delta'(x - x_*)$,

$$(1, e)_{h1} = 0 \qquad (x^k, e)_{h1} = -k(x_*)^{k-1} \quad k = 1, \ldots, 4,$$

and is twice continuously differentiable with respect to the source location x_*, i.e., satisfies continuity conditions corresponding to (24).

5 Estimating initial source parameters

Figure 1 shows an example of contour levels of $\mathcal{X}$ in two planes of the 11-dimensional parameter space, where the remaining nine parameters are held at their minimizing values. This example is taken from the layer over half space problem called LOH.1 that is described in section 6. The minimum is clearly visible at $x_* = y_* = 15000$ and $z_* = 2000$. Gradient based minimization algorithms assume that the objective function is close to quadratic in parameter space. Figure 1 shows that this assumption only holds close to the minima, and several local minima can be distinguished in the figure. To make the minimization algorithm converge to the global minima, it follows that the initial parameter guess must be fairly accurate. We proceed by describing an approach for establishing initial parameter values for the source estimation problem.

5.1 Initial estimate for the source location and start time

Our initial estimate for the source location is based on first arrival times. Assume that the first wave arrives at time t_r at receiver location (x_r, y_r, z_r). If the material has homogeneous properties with compressional wave velocity c_p, the travel time from source location (x_*, y_*, z_*) to receiver 'r' satisfies

$$\hat{T}_r(x_*, y_*, z_*) = \frac{1}{c_p}\sqrt{(x_r - x_*)^2 + (y_r - y_*)^2 + (z_r - z_*)^2}.$$

The source starting time, t_*, is related to the first arrival time, t_r, through $\hat{T}_r(x_*, y_*, z_*) + t_* = t_r$. Hence, we consider

$$I_r(x_*, y_*, z_*, t_*) := \hat{T}_r(x_*, y_*, z_*) + t_* - t_r = 0, \quad r = 1, \ldots, R. \tag{37}$$

Each receiever results in one equation for the four unknowns (x_*, y_*, z_*, t_*). To determine the unknowns, we need at least four receievers. Usually, we have more than four receivers and (37) becomes an overdetermined system, which can be solved in the least squares sense using the Gauss-Newton method.

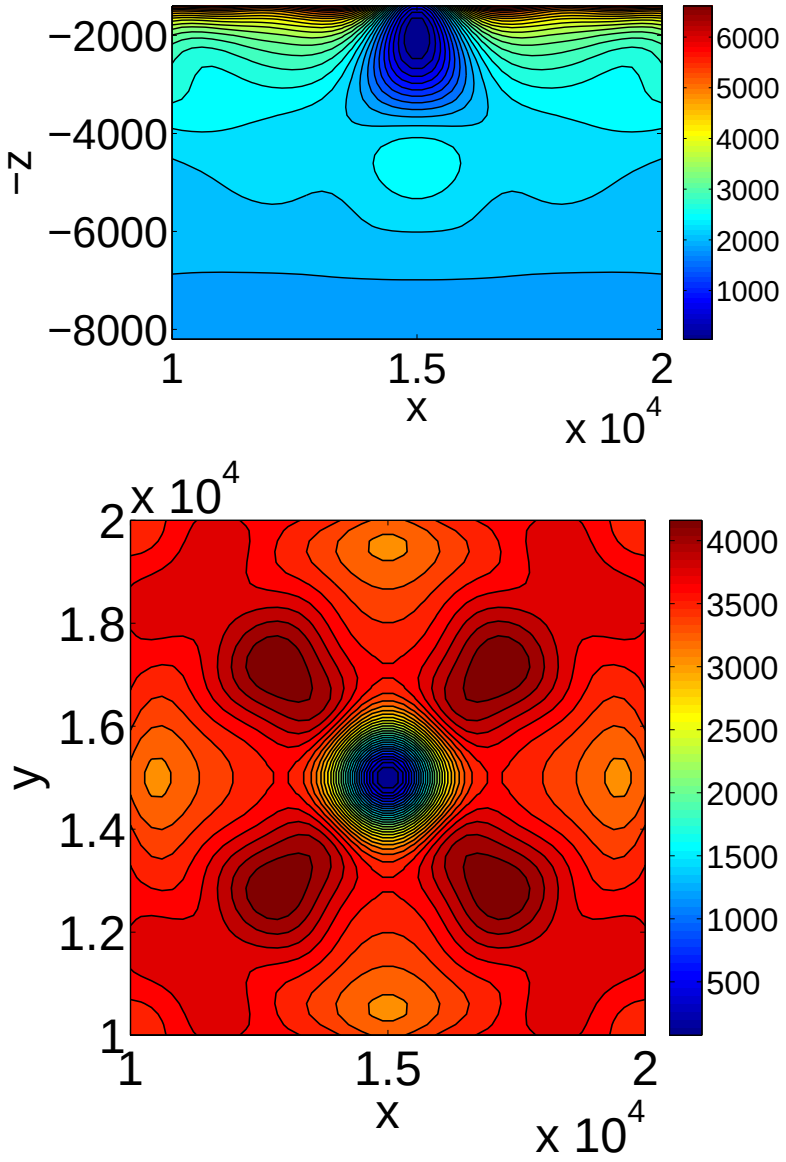

Figure 1: Contour plots of $\mathcal{X}$ as function of the source location for the LOH.1 problem. Top: $\mathcal{X}(x_*, z_*)$ for $y_* = 15000$. Bottom: $\mathcal{X}(x_*, y_*)$ for $z_* = 2000$. All other parameters are held at their correct values.

The first arrival time at receiver r is defined as the smallest value of t_m for which

$$|u^m_{\mathbf{i}_r}| > \eta \max_n |u^n_{\mathbf{i}_r}| \text{ or } |v^m_{\mathbf{i}_r}| > \eta \max_n |v^n_{\mathbf{i}_r}| \text{or } |w^m_{\mathbf{i}_r}| > \eta \max_n |w^n_{\mathbf{i}_r}|, \quad 0 < \eta \ll 1, \tag{38}$$

i.e., the earliest time for which the amplitude of any component of the signal reaches a fraction of its peak value. In the numerical experiments below we use $\eta = 10^{-6}$.

If the material is not homogeneous, the above procedure can still be applied if c_p is replaced by an appropriate average of the compressional wave speed. However, we have found that such an approach can be sensitive to the value of c_p. In some cases it even makes the Gauss-Newton iteration diverge. For heterogeneous materials, a better approach is to estimate the initial source position by ray tracing, which takes the variable material properties into account. We here describe the approach for a layered material model. Consider a material with piecewise constant compressional wave speed in m horizontal layers, depending only on the depth below the free surface,

$$c_p(z) = \begin{cases} c_1, & 0 \le z < z_1, \\ c_2, & z_1 \le z < z_2, \\ & \cdots \\ c_m, & z_{m-1} \le z \le z_{max}. \end{cases}$$

Assume that the source is located at $(x_*,\ y_*,\ z_*)$ with $z_{n-1} < z_* < z_n$, and that the receiver is located at $(x_r,\ y_r,\ 0)$. The piecewise linear path between source and receiver with break points at (x_i, y_i, z_i), $i = 1, \ldots, n-1$ has the travel time

$$T_r = \sum_{i=1}^{n} \frac{\xi_i}{c_i}, \quad \xi_i = \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 + (z_i - z_{i-1})^2},$$

where we set the start and end points of the ray to $(x_0, y_0, z_0) = (x_r, y_r, 0)$ and $(x_n, y_n, z_n) = (x_*, y_*, z_*)$, respectively. The travel time is a function of the locations of the break points (x_i, y_i, z_i), $i = 1, \ldots, n-1$. Since z_i are fixed, the ray from source to receiver is determined by the values of (x_i, y_i) that minimize T_r.

At the minimum, $\partial T_r/\partial x_i = 0$ and $\partial T_r/\partial y_i = 0$, for $i = 1, 2, \ldots, n-1$. This results in the system

$$A(\mathbf{x}, \mathbf{y})\mathbf{x} = \mathbf{r}^{(1)}, \quad A(\mathbf{x}, \mathbf{y})\mathbf{y} = \mathbf{r}^{(2)}, \tag{39}$$

where $\mathbf{x} = (x_1, \ldots, x_{n-1})$, $\mathbf{y} = (y_1, \ldots, y_{n-1})$, and the matrix A is tridiagonal. Note that equation (39) is non-linear, because A depends on $\mathbf{x}$ and $\mathbf{y}$. We have found that (39) can be solved by the fix point iteration

$$A(\mathbf{x}^k, \mathbf{y}^k)\mathbf{x}^{k+1} = \mathbf{r}^{(1)}, \quad A(\mathbf{x}^k, \mathbf{y}^k)\mathbf{y}^{k+1} = \mathbf{r}^{(2)}, \quad k = 0, 1, \ldots.$$

The iteration converges quickly, at least for cases where the number of layers is moderate.

By solving the above minimization problem, we can calculate the minimum travel time from the source to receiver 'r', which we denote by $\hat{T}_r$. Clearly, it is a function of the source location, and similar to the case with a homogeneous material, we have $\hat{T}_r = \hat{T}_r(x_*, y_*, z_*)$.

As before, we can determine the source location by solving (37) for the unknowns (x_*, y_*, z_*, t_*). We assume $R > 4$, which makes (37) an overdetermined system. The Jacobian of (37) is needed by the Gauss-Newton method. Here we approximate the derivative of I_r with respect to the source location by numerical differentiation, i.e.,

$$\frac{\partial I_r}{\partial x_*} \approx \frac{\hat{T}_r(x_* + \tilde{h}, y_*, z_*) - \hat{T}_r(x_*, y_*, z_*)}{\tilde{h}},$$

for a small fixed number $\tilde{h}$. The y_*- and z_*-derivatives are computed similarly. Thus when solving (37) by the Gauss-Newton method, we must solve (39) four times per iteration.

5.2 Estimating the source frequency

It has turned out to be difficult to automatically estimate the source frequency, ω_0. For this reason, we require an initial guess for ω_0 to be provided by the user. However, in practice this might not be a serious problem, because in realistic applications the observed ground motions must be filtered in time to remove waves that can not be resolved on the computational grid. This is a preprocessing step that is performed before the optimization is started. The corner frequency of the filter is then related to the effective source frequency.

5.3 Initial estimate for the moment tensor

Once initial estimates for the source location, frequency, and starting time are established, we can use the linearity of the elastic wave equation to estimate the matrix $\mathcal{M}$ in the source term (4). Let $\mathbf{u}^{(xx)}$, $\mathbf{u}^{(xy)}$, $\mathbf{u}^{(xz)}$, $\mathbf{u}^{(yy)}$, $\mathbf{u}^{(yz)}$, and $\mathbf{u}^{(zz)}$ denote solutions of the elastic wave equation with the matrix $\mathcal{M}$ set to

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \quad \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \quad \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

respectively. The solution for a general $\mathcal{M}$ is then obtained as the linear combination

$$\mathbf{w}(m_{xx}, m_{xy}, m_{xz}, m_{yy}, m_{yz}, m_{zz}) := \\ m_{xx}\mathbf{u}^{(xx)} + m_{xy}\mathbf{u}^{(xy)} + m_{xz}\mathbf{u}^{(xz)} + m_{yy}\mathbf{u}^{(yy)} + m_{yz}\mathbf{u}^{(yz)} + m_{zz}\mathbf{u}^{(zz)}.$$

The elements of $\mathcal{M}$ are determined by minimizing the wave form misfit

$$\mathcal{X} = \frac{1}{2} \sum_{r=1}^{R} \sum_{n=0}^{M-1} s(t_n) \left| \mathbf{w}_{\mathbf{i}_r}^n(m_{xx}, m_{xy}, m_{xz}, m_{yy}, m_{yz}, m_{zz}) - \mathbf{d}_{\mathbf{i}_r}^n \right|^2. \tag{40}$$

Because $\mathbf{w}$ is linear in m_{ij}, $\mathcal{X}$ is a quadratic function of m_{ij}. Its minimum can be computed directly by solving the 6×6 linear system $\partial \mathcal{X} / \partial m_{ij} = 0$.

6 Numerical experiments

To verify our implementation and gain understanding of the performance of the suggested approach, we conduct numerical experiments on synthetic data. In this approach we first run the forward problem once with a given source and use the resulting time histories at the receiver stations as 'measured' data. In this way, the exact solution is known, and we can easily assess the convergence properties of the minimization algorithm.

A standard test problem for elastic wave modeling is the layer over halfspace problem called LOH.1, [2]. In this test, a point moment tensor source with a Gaussian time function is applied in a layered isotropic elastic material. The Gaussian time function,

$$g(t; t_0, \omega_0) = \frac{\omega_0}{\sqrt{2\pi}} e^{-\omega_0^2 (t-t_0)^2/2},$$

is parameterized by the frequency ω_0 and the center time t_0. The material velocities are $c_p = 4000$ and $c_s = 2000$ in a top layer extending over $0 \leq z \leq 1000$, with $c_p = 6000$ and $c_s = 3464$ for $z > 1000$. The densities are $\rho = 2600$ in the top layer and $\rho = 2700$ for $z > 1000$. The computational domain in the standard LOH.1 problem is a box of size $30000 \times 30000 \times 17000$. In order to make the computations run faster, we reduce the depth of the computational domain from 17000 to 8500, but keep the thickness of the top layer unchanged. All computations use the grid spacing $h = 120$ and the wave equation is integrated to time $T = 9$. The spatial grid has 4.5 million points.

In the following numerical experiments, the 'exact' source is located at $x_* = y_* = 15000$, $z_* = 2000$, with moments $m_{xy} = 10^{18}$, $m_{xx} = m_{xz} = m_{yy} = m_{yz} = m_{zz} = 0$. The Gaussian time function has $t_0 = 1.45$ and $\omega_0 = 6.0$. To allow for a coarser grid spacing than what normally is used when solving the LOH.1 problem, we have reduced the value for the frequency parameter ω_0 to give a similar resolution in terms of grid points per wavelength. The solution is recorded at 25 receiver stations placed on a 5×5 grid, at $x_{j,k} = 9000 + (j-1) * 3000$, $y_{j,k} = 9000 + (k-1) * 3000$, and $z_{j,k} = 0$, for $j = 1, \ldots, 5$, $k = 1, \ldots, 5$.

The center time t_0 in the Gaussian time function follows from the source start time t_* as $t_0 = t_* + t_\delta$, where the rise time t_δ satisfies

$$\eta = \frac{\omega_0}{\sqrt{2\pi}} e^{-(\omega_0 t_\delta)^2/2} \quad \Rightarrow \quad t_\delta = \frac{1}{\omega_0} \sqrt{-2 \log((\eta\sqrt{2\pi})/\omega_0)}.$$

Here, $0 < \eta \ll 1$ is the same constant that is used in (38) for estimating the first arrival time.

The automated initial guess described in Section 5, with the user choice $\omega_0 = 6.3$, resulted in the estimated source position $x_* = 14980.8$, $y_* = 15039.5$, $z_* = 2352.73$, with moments $m_{xx} = 3.754 \times 10^{14}$, $m_{xy} = 9.622 \times 10^{17}$, $m_{xz} = -8.937 \times 10^{15}$, $m_{yy} = 3.758 \times 10^{14}$, $m_{yz} = 4.351 \times 10^{15}$, $m_{zz} = -5.313 \times 10^{12}$, and $t_0 = 1.358$. This estimate was computed by solving (37) in the two-layer material above. It is a sufficiently good approximation to make the Fletcher-Reeves algorithm converge to the exact minimum.

In our implementation of the source inversion algorithm, the user is given the choice to input an initial parameter guess directly, or to have the solver estimate it. When a

fairly accurate location of the source is known in some other way, it saves computational time to use it directly. The computational cost of the automatic estimate is somewhat high because the computation of the moment tensor components requires six elastic wave equations to be solved, see Section 5.3. In some of the numerical experiments below, we use the following initial parameter values:

$$\begin{aligned} x_* = 16000,\ \ y_* = 14000,\ \ z_* = 2200,\ \ m_{xy} = 1.2 \cdot 10^{18}, \\ m_{xx} = m_{xz} = m_{yy} = m_{yz} = m_{zz} = 0,\ \ t_0 = 1.54,\ \ \omega_0 = 6.3. \end{aligned} \tag{41}$$

As long as the initial parameter guess is reasonably close to the actual minimum, our practical experience is that the number of iterations required to reach convergence is not sensitive to this choice.

6.1 Choosing the preconditioner

The sizes of the parameters in the source estimation probem span many orders of magnitude. In SI-units, $\mathbf{x}_*$ is of the order $\mathcal{O}(10^4)$, the moment tensor components m_{xx}, $m_{xy}, \ldots$ are of the order $\mathcal{O}(10^{15}) - \mathcal{O}(10^{18})$. The parameters t_0 and ω_0 are both of the order $\mathcal{O}(1) - \mathcal{O}(10)$. Because there is such a large difference in size between the smallest and largest parameter values, the original minimization problem is very poorly scaled and the condition number of the Hessian is a very large.

For optimal convergence of the Fletcher-Reeves algorithm, the parameters should be scaled such that the Hessian at the solution has condition number one. The change of parameters $\hat{\mathbf{p}} = S\mathbf{p}$ gives the scaled Hessian $\hat{H}_* = (S^{-1})^T H_* S^{-1}$, i.e., the scaling corresponding to $\hat{H}_* = I$ satisfies

$$S^T S = H_*. \tag{42}$$

Hence, S could be computed by a Cholesky factorization of H_*. However, H_* is in general not computable because it requires the solution of the minimization problem to be known. Instead we can use a Cholesky factorization of the Hessian at the initial parameter guess. Since this scaling is not optimal and the implementation of the scaled algorithm is more straight forward when S is diagonal, we restrict us to this case. As we shall see, a significant reduction of the condition number of the Hessian can still be acheived. When S is diagonal, (42) can not be satisfied exactly. Instead we minimize the residual, $||H_* - S^2||_F$, which gives $S_{jj} = \sqrt{H_{jj}}$, $j = 1, \ldots, Q$, i.e., the scaling matrix should equal the square root of the diagonal of the Hessian. The Hessian at the minimum is positive definite, which implies that the diagonal elements of H_* are positive. Since H_* is not known until the minimization problem has been solved, we define S as the square root of the diagonal of the Hessian evaluated at the initial guess. One difficulty with this definition is that there is no guarantee that the Hessian is positive definite at the initial guess. If there are negative diagonal elements in the Hessian, we instead use the square root of the diagonal elements of the matrix H_1, see (18), evaluated at the initial guess. It is obvious from (18) that H_1 always have non-negative diagonal elements. This approach has turned out to work well in practice.

The computation of the Hessian, which is described in Section 3.1, requires the elastic wave equation to be solved 11 times. However, this computation only needs to be done once, before the Fletcher-Reeves iteration starts.

6.2 Condition number of the scaled Hessian

Table 1 shows the influence of different scaling matrices for the LOH.1 source inversion problem. The bottom row shows the condition number of the Hessian at the exact minimum, scaled by the given S. Here, the condition number was computed by the Matlab function `cond`. The diagonal variable transformation $\hat{\mathbf{p}} = S\mathbf{p}$ implies that the inverse of the diagonal elements of the scaling matrix correspond to reference sizes of the parameters. However, only their relative sizes matter because multiplying S by a constant factor does not change the condition number of the scaled Hessian.

The unscaled Hessian has condition number $cond(H_*) = 4.73 \cdot 10^{38}$. The second column of Table 1 shows the scaling obtained as the square root of the diagonal elements of the Hessian, evaluated at the initial parameter guess (41). Not all diagonal elements of the Hessian are positive at this point in parameter space. For this reason, only the first part of the Hessian, H_1, as described in Section 6.1 was used in this scaling. It is interesting to note that the scaling obtained from the square root of the diagonal of the Hessian at the minimum, shown in column three, leads to a slightly larger condition number. As was mentioned above, the Hessian at the minimum is in general not computable because this scaling assumes that the solution of the minimization problem is known. The fourth column, labeled ref. sizes 1, shows the scaling based on estimated sizes of the parameters. These numbers were based on the size of the domain, which is in the 10's of kilometers, and the fact that we know that t_0 is of order $\mathcal{O}(1)$, $\omega_0 = \mathcal{O}(10)$, and the moment tensor components are of the order $\mathcal{O}(10^{18})$. Table 1 shows that scalings based on the diagonal elements of the Hessian give significantly smaller condition numbers compared with the unscaled case and ref. sizes 1. After inspecting the Hessian based scalings, we modified the reference size scaling to instead be 10^3 for the length scale, 10^{18} for the moment scale, 0.1 for the time scale, and 1 for the frequency scale. This scaling, given as ref. sizes 2 in Table 1, gave the much improved condition number 80.4. The last column of Table 1 shows a scaling that is in between ref. sizes 1 and ref. sizes 2, labeled ref. sizes 3. It lead to a condition number close to that of ref. sizes 1, indicating how sensitive the condition number is to the scaling matrix. Hence, even though it is possible to design a favorable parameter scaling by order of magnitude arguments, it is very difficult to find the optimal values. We conclude that using the Hessian for scaling the parameters is a much more reliable way to acheive a small condition number.

6.3 Convergence rates

Figure 2 shows convergence properties of the Fletcher-Reeves algorithm for different scaling matrices. These computations were run for up to a maximum of 10 restarts ($m = 10$ in Algorithm 3), with each restart cycle consisting of 11 inner iterations. The iteration is terminated when the maximum norm of the scaled gradient of the misfit becomes smaller

	Hes., guess	Hes., exact	Ref. sizes 1	Ref. sizes 2	Ref. sizes 3
$1/s_{1,1}$ (x_*)	10.8	11.6	$1.00 \cdot 10^4$	$1.00 \cdot 10^3$	$5.00 \cdot 10^3$
$1/s_{2,2}$ (y_*)	10.8	11.6	$1.00 \cdot 10^4$	$1.00 \cdot 10^3$	$5.00 \cdot 10^3$
$1/s_{3,3}$ (z_*)	12.2	20.6	$1.00 \cdot 10^4$	$1.00 \cdot 10^3$	$5.00 \cdot 10^3$
$1/s_{4,4}$ (m_{xx})	$2.68 \cdot 10^{16}$	$2.70 \cdot 10^{16}$	$1.00 \cdot 10^{18}$	$1.00 \cdot 10^{18}$	$1.00 \cdot 10^{18}$
$1/s_{5,5}$ (m_{xy})	$1.68 \cdot 10^{16}$	$1.65 \cdot 10^{16}$	$1.00 \cdot 10^{18}$	$1.00 \cdot 10^{18}$	$1.00 \cdot 10^{18}$
$1/s_{6,6}$ (m_{xz})	$1.39 \cdot 10^{16}$	$1.29 \cdot 10^{16}$	$1.00 \cdot 10^{18}$	$1.00 \cdot 10^{18}$	$1.00 \cdot 10^{18}$
$1/s_{7,7}$ (m_{yy})	$2.67 \cdot 10^{16}$	$2.70 \cdot 10^{16}$	$1.00 \cdot 10^{18}$	$1.00 \cdot 10^{18}$	$1.00 \cdot 10^{18}$
$1/s_{8,8}$ (m_{yz})	$1.39 \cdot 10^{16}$	$1.29 \cdot 10^{16}$	$1.00 \cdot 10^{18}$	$1.00 \cdot 10^{18}$	$1.00 \cdot 10^{18}$
$1/s_{9,9}$ (m_{zz})	$2.28 \cdot 10^{16}$	$1.85 \cdot 10^{16}$	$1.00 \cdot 10^{18}$	$1.00 \cdot 10^{18}$	$1.00 \cdot 10^{18}$
$1/s_{10,10}$ (t_0)	$2.13 \cdot 10^{-3}$	$2.55 \cdot 10^{-3}$	1.00	0.10	0.50
$1/s_{11,11}$ (ω_0)	$5.65 \cdot 10^{-2}$	$6.24 \cdot 10^{-2}$	10.0	1.00	5.00
$cond(S^{-1}H_*S^{-1})$	28.1	31.8	$6.18 \cdot 10^3$	80.4	$1.55 \cdot 10^3$

Table 1: *Scaling factors and their influence on the condition number of the scaled Hessian. The condition number of the unscaled Hessian is* $2.77 \cdot 10^{38}$

than 10^{-12}, or when the maximum number of iterations is reached. The magenta curve shows the convergence history when S is taken as the square root of the diagonal elements of H_*. Using the square root of the diagonal elements of the Hessian at the initial guess, shown by the red curve in Figure 2, gives convergence in almost the same number of iterations. The results plotted in cyan and blue colors were obtained with scalings corresponding to the cases "ref. sizes 1" and "ref. sizes 2" in Table 1. The improvement in convergence rate when switching from "ref. sizes 1" to "ref. sizes 2" is remarkable, and prompted us to try the intermediate scaling labeled "ref. sizes 3", shown in green color.

The Hessian based scalings always perform well, and the solution is obtained in 40-50 iterations. The reference size scalings can be made almost as efficient, but are very sensitive to the exact values in the scaling matrix. The convergence rate of the unscaled method, shown in black, is very slow and should not be used in practical computations. Figure 3 displays the evolution of the source parameters during the iterations with the scaling computed from the Hessian at the initial guess. The left figure shows the source position x_*, y_*, z_* vs. the number of iterations. y_* is offset by 1000 to distinguish it from x_* and z_* is offset by 10000 to make it fit into the same plot. The circles to the right of the curves indicate the exact value of the parameter. Similarly, the middle subplot of Fig. 3 show the evolution of the six components of the matrix $\mathcal{M}$, and the right subplot shows the time shift and the frequency with an offset. It is clear that already after around 33 iterations (2 restarts with the Fletcher-Reeves algorithm), all parameter values have converged in "picture" norm.

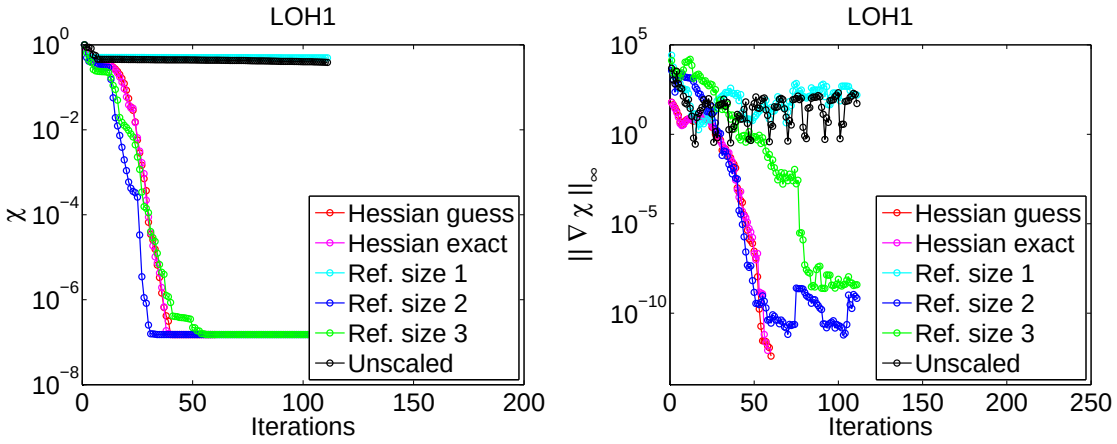

Figure 2: Convergence of the misfit (left) and the maximum norm of the scaled gradient (right) for the different scalings given in Table 1.

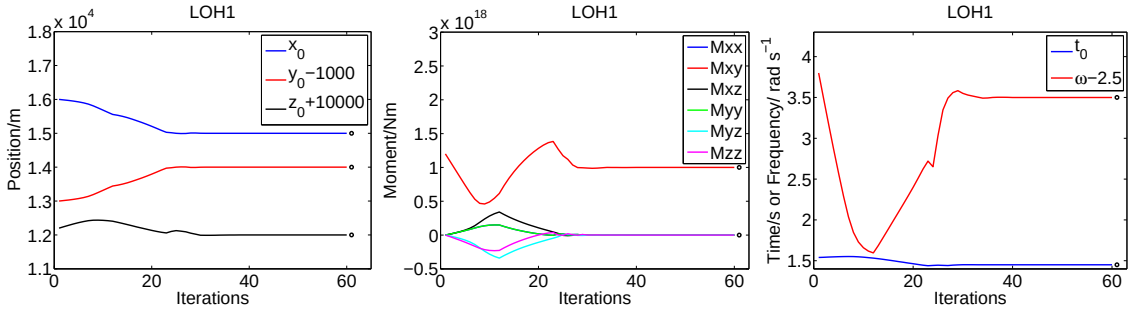

Figure 3: Convergence of source parameters. Location (left), moment tensor components (middle), and time shift and frequency (right).

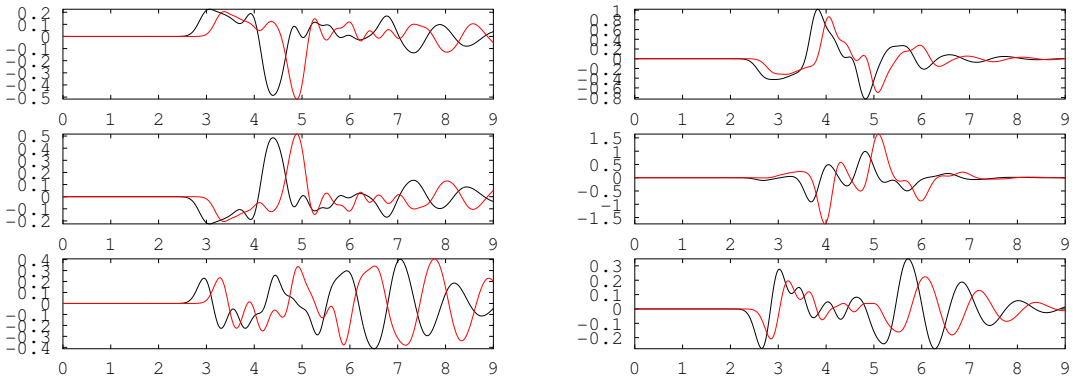

Figure 4: The (u, v, w) components of the solution at the receivers $(x, y, z) = (9000, 21000, 0)$ (left) and $(x, y, z) = (9000, 12000, 0)$ (right) vs. time. Curves in red are initial guesses and curves in black are converged solutions.

Figure 4 gives another illustration of the convergence process. The recorded ground motions at the receiver stations, $(x, y, z) = (9000, 21000, 0)$ and $(x, y, z) = (9000, 12000, 0)$ are displayed as functions of time. The red curves were computed with the initial parameter values (41). The curves plotted in black are the solutions at the minimum. Since this example uses synthetic data, the converged results are identical to the observations, modulus roundoff errors.

6.4 Computational cost

Each iteration of the Fletcher-Reeves algorithm requires a minimum of three elastic wave equations and one adjoint wave equation to be solved. The minumum number of wave equation solves holds if the line search algorithm accepts the initial step length, which is usually the case. When the line search algorithm needs to perform step length reduction (backtracking), each reduction incurs one additional solve of an elastic wave equation.

The number of wave equation solves are counted as follows. In each iteratation, the computation of the misfit, its gradient, and part H_2 of the Hessian can be obtained by solving one elastic and one adjoint wave equation. The computation of the step length (13) adds one more elastic wave equation to be solved, and the test for acceptance at the initial step in the line search algorithm requires the solution of yet another elastic wave equation.

Figure 3 shows that a highly accurate solution can be obtained after around 40 iterations. Since the adjoint wave equation needs the same amount of computational work as solving the elastic wave equation, this corresponds to approximately 160 solutions of the elastic wave equation. The total computation times for the 60 iterations shown in Figure 3 was about 15 minutes using 256 cores of a Linux cluster of Intel Xenon processors.

7 Conclusions

We have presented an algorithm for estimating the seismic source parameters from recorded time dependent motions at a number of receiver stations. The solution of this inverse problem is computed by minimizing the full waveform misfit using a non-linear conjugate gradient method. The key freatures of the proposed technique are an adjoint discretization of the fourth order accurate method in [12], a source discretization that leads to a twice continuously differentiable misfit function, and a parameter scaling that makes the minimization problem well conditioned. Numerical experiments with the LOH.1 problem shows very good convergence rates of the proposed algorithm.

Several practical problems must be solved before we can apply our approach to estimate source parameters in realistic seismic events. Seismographic recordings must be deconvolved and band-pass filtered to compensate for instrument response characteristics. Furthermore, additional filtering of the recordings is often needed to remove frequencies that can not be resolved on the computational grid. We are currently faced with the difficulty of incorporating the filters into the source inversion problem. Instead of applying filters to the computed solution, we plan to filter the source time function. Because the material properties do not depend on time, the elastic wave equation describes a linear time-invariant system. Filtering the solution is therefore equivalent to using a filtered source time function. As a result, the minimization problem will depend on only 10 parameters, since the frequency parameter is fixed by the filter.

Another interesting extension of the current approach is the inverse problem for estimating the material wave speeds and density. One additional difficulty with this problem is to find a suitable parameterization of the material. To limit the dimensionality of parameter space, we expect that some degree of smoothness must be imposed on the material model, perhaps by using piecewise smooth basis functions. Preliminary computations in two space dimensions with a very simple material parameterization have shown encouraging results.

A Proof of Theorem 1

Proof. Expand the predictor into the corrector in Algorithm 1. Then rewrite the expression as

$$\rho\frac{\mathbf{u}^{n+1}-2\mathbf{u}^n+\mathbf{u}^{n-1}}{\Delta_t^2} = \mathbf{L}_h(\mathbf{u}^n)+\mathbf{F}(t_n)+\frac{\Delta_t^2}{12}\big(\mathbf{L}_h(\mathbf{v}^n)+\mathbf{F}_{tt}(t_n)\big)+\mathbf{S}_G(\mathbf{u}^n-\mathbf{u}^{n-1}). \quad (43)$$

Next, take the scalar product of (43) and $\boldsymbol{\kappa}^n$, and sum over time to obtain

$$\sum_{n=0}^{M-1}(\boldsymbol{\kappa}^n,\rho\frac{\mathbf{u}^{n+1}-2\mathbf{u}^n+\mathbf{u}^{n-1}}{\Delta_t^2})_h = \sum_{n=0}^{M-1}(\boldsymbol{\kappa}^n,\mathbf{L}_h(\mathbf{u}^n))_h$$
$$+\sum_{n=0}^{M-1}(\boldsymbol{\kappa}^n,\mathbf{F}(t_n)+\frac{\Delta_t^2}{12}\mathbf{F}_{tt}(t_n))_h+\frac{\Delta_t^2}{12}\sum_{n=0}^{M-1}(\boldsymbol{\kappa}^n,\mathbf{L}_h(\mathbf{v}^n))_h+\sum_{n=0}^{M-1}(\boldsymbol{\kappa}^n,\mathbf{S}_G(\mathbf{u}^n-\mathbf{u}^{n-1}))_h. \quad (44)$$

The sum on the left hand side of (44) can be rewritten

$$\sum_{n=0}^{M-1}(\boldsymbol{\kappa}^n,\rho\frac{\mathbf{u}^{n+1}-2\mathbf{u}^n+\mathbf{u}^{n-1}}{\Delta_t^2})_h=\sum_{n=0}^{M-1}(\rho\frac{\boldsymbol{\kappa}^{n+1}-2\boldsymbol{\kappa}^n+\boldsymbol{\kappa}^{n-1}}{\Delta_t^2},\mathbf{u}^n)_h$$
$$+\frac{1}{\Delta_t^2}\left((\rho\mathbf{u}^{-1},\boldsymbol{\kappa}^0)_h-(\rho\mathbf{u}^0,\boldsymbol{\kappa}^{-1})_h+(\rho\mathbf{u}^M,\boldsymbol{\kappa}^{M-1})_h-(\rho\mathbf{u}^{M-1},\boldsymbol{\kappa}^M)_h\right), \quad (45)$$

where the initial data $\mathbf{u}^0=\mathbf{u}^{-1}=\boldsymbol{\kappa}^M=\boldsymbol{\kappa}^{M-1}=\mathbf{0}$ make

$$\sum_{n=0}^{M-1}(\boldsymbol{\kappa}^n,\rho\frac{\mathbf{u}^{n+1}-2\mathbf{u}^n+\mathbf{u}^{n-1}}{\Delta_t^2})_h=\sum_{n=0}^{M-1}(\rho\frac{\boldsymbol{\kappa}^{n+1}-2\boldsymbol{\kappa}^n+\boldsymbol{\kappa}^{n-1}}{\Delta_t^2},\mathbf{u}^n)_h.$$

The first sum on the right hand side of (44) is treated by the self-adjoint property,

$$\sum_{n=0}^{M-1}(\boldsymbol{\kappa}^n,\mathbf{L}_h(\mathbf{u}^n))_h=\sum_{n=0}^{M-1}(\mathbf{L}_h(\boldsymbol{\kappa}^n),\mathbf{u}^n)_h$$

The third sum of the right hand side of (44) can be rewritten

$$(\boldsymbol{\kappa}^n,\mathbf{L}_h(\mathbf{v}^n))_h=(\mathbf{L}_h(\boldsymbol{\kappa}^n),\mathbf{v}^n)_h=(\mathbf{L}_h(\boldsymbol{\kappa}^n),\frac{1}{\rho}(\mathbf{L}_h(\mathbf{u}^n)+\mathbf{F}(t_n)))_h=$$
$$(\mathbf{L}_h(\boldsymbol{\kappa}^n),\frac{1}{\rho}\mathbf{L}_h(\mathbf{u}^n))_h+(\mathbf{L}_h(\boldsymbol{\kappa}^n),\frac{1}{\rho}\mathbf{F}(t_n))_h$$
$$=(\mathbf{L}_h(\frac{1}{\rho}\mathbf{L}_h(\boldsymbol{\kappa}^n)),\mathbf{u}^n)_h+(\mathbf{L}_h(\boldsymbol{\kappa}^n),\frac{1}{\rho}\mathbf{F}(t_n))_h=(\mathbf{L}_h(\boldsymbol{\zeta}^n),\mathbf{u}^n)_h+(\boldsymbol{\zeta}^n,\mathbf{F}(t_n))_h \quad (46)$$

The supergrid damping term is rewritten

$$\sum_{n=0}^{M-1}(\boldsymbol{\kappa}^n,\mathbf{S}_G(\mathbf{u}^n-\mathbf{u}^{n-1}))_h=\sum_{n=0}^{M-1}(\boldsymbol{\kappa}^n,\mathbf{S}_G(\mathbf{u}^n))_h$$
$$-\sum_{n=0}^{M-1}(\boldsymbol{\kappa}^{n+1},\mathbf{S}_G(\mathbf{u}^n))_h-(\boldsymbol{\kappa}^0,\mathbf{S}_G(\mathbf{u}^{-1}))_h+(\boldsymbol{\kappa}^M,\mathbf{S}_G(\mathbf{u}^{M-1}))_h. \quad (47)$$

The boundary terms are zero because of the initial data $\mathbf{u}^{-1}=\boldsymbol{\kappa}^M=0$, and we use the symmetry (8) to obtain

$$\sum_{n=0}^{M-1}(\boldsymbol{\kappa}^n,\mathbf{S}_G(\mathbf{u}^n-\mathbf{u}^{n-1}))_h=\sum_{n=0}^{M-1}(\boldsymbol{\kappa}^n-\boldsymbol{\kappa}^{n+1},\mathbf{S}_G(\mathbf{u}^n))_h=-\sum_{n=0}^{M-1}(\mathbf{S}_G(\boldsymbol{\kappa}^{n+1}-\boldsymbol{\kappa}^n),\mathbf{u}^n)_h$$

Collecting terms gives that (44) is equivalent to

$$\sum_{n=0}^{M-1}(\rho\frac{\boldsymbol{\kappa}^{n+1}-2\boldsymbol{\kappa}^n+\boldsymbol{\kappa}^{n-1}}{\Delta_t^2},\mathbf{u}^n)_h=\sum_{n=0}^{M-1}(\mathbf{L}_h(\boldsymbol{\kappa}^n),\mathbf{u}^n)_h+\sum_{n=0}^{M-1}(\boldsymbol{\kappa}^n,\mathbf{F}(t_n)+\frac{\Delta_t^2}{12}\mathbf{F}_{tt}(t_n))_h$$
$$+\frac{\Delta_t^2}{12}\sum_{n=0}^{M-1}(\mathbf{L}_h(\boldsymbol{\zeta}^n),\mathbf{u}^n)_h+\frac{\Delta_t^2}{12}\sum_{n=0}^{M-1}(\boldsymbol{\zeta}^n,\mathbf{F}(t_n))_h-\sum_{n=0}^{M-1}(\mathbf{S}_G(\boldsymbol{\kappa}^{n+1}-\boldsymbol{\kappa}^n),\mathbf{u}^n)_h. \quad (48)$$

Expanding the predictor (10) into the corrector (11) gives

$$\rho\frac{\boldsymbol{\kappa}^{n+1}-2\boldsymbol{\kappa}^{n}+\boldsymbol{\kappa}^{n-1}}{\Delta_t^2}=\mathbf{L}_h(\boldsymbol{\kappa}^n)+\mathbf{G}(t_n)+\frac{\Delta_t^2}{12}\mathbf{L}_h(\boldsymbol{\zeta}^n)-\mathbf{S}_G(\boldsymbol{\kappa}^{n+1}-\boldsymbol{\kappa}^n). \qquad (49)$$

It is now straightforward to see, by comparing (48) and (49), that the identity (12) follows. □

References

[1] A.A.Allam and Y.Ben-Zion. Seismic velocity structures in the southern California plate-boundary environment from double-difference tomography. *Geophys. J. Int.*, 190:1181–1196, 2012.

[2] S. M. Day, J. Bielak, D. Dreger, S. Larsen, R. Graves, A. Pitarka, and K. B. Olsen. Test of 3D elastodynamic codes: Lifelines project task 1A01. Technical report, Pacific Earthquake Engineering Center, 2001.

[3] J.E. Dennis, Jr. and R.B.Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice-Hall, 1983.

[4] E.Haber. Quasi-Newton methods for large-scale electromagnetic inverse problems. *Inverse Problems*, 21:305–323, 2005.

[5] Y. Kim, Q. Liu, and J. Tromp. Adjoint centroid-moment tensor inversions. *Geophys. J. Int.*, 186(1):264–278, 2011.

[6] L.Eisner and R.W.Clayton. A reciprocity method for multiple-source simulations. *Bull. Seism. Soc. Am.*, 91:553–560, 2001.

[7] D. G. Luenberger. *Introduction to Linear and Nonlinear Programming*. Addison-Wesley, 1973.

[8] S. Nilsson, N. A. Petersson, B. Sjögreen, and H.-O. Kreiss. Stable difference approximations for the elastic wave equation in second order formulation. *SIAM J. Numer. Anal.*, 45:1902–1936, 2007.

[9] N. A. Petersson and B. Sjögreen. Stable grid refinement and singular source discretization for seismic wave simulations. *Comm. Comput. Phys.*, 8(5):1074–1110, November 2010.

[10] R.-E. Plessix and W.A.Mulder. Resistivity imaging with controlled-source electromagnetic data: depth and data weighting. *Inverse Problems*, 24:034012, 2008.

[11] P.Liu, S.Hartzell, and W.Stephenson. Non-linear multiparameter inversion using a hybrid global search algorithm: applications in reflection seismology. *Geophys. J. Int.*, 122:991–1000, 1995.

[12] B. Sjögreen and N. A. Petersson. A fourth order accurate finite difference scheme for the elastic wave equation in second order formulation. *J. Scient. Comput.*, 52:17–48, 2012.

[13] C. Tape, Q. Liu, and J. Tromp. Finite-frequency tomography using adjoint methods - Methodology and examples using membrane surface waves. *Geophys. J. Int.*, 168:1105–1129, 2007.

[14] T.G.Kolda, R.M.Lewis, and V.Torczon. Optimization by direct search: New perspectives on some classical and modern methods. *SIAM Rev.*, 45:385–482, 2003.

[15] J. Tromp, C. Tape, and Q. Liu. Seismic tomography, adjoint mehtods, time reversal and banana-doughnut kernels. *Geophys. J. Int.*, 160:195–216, 2005.

[16] Y.H.Kim, Q. Liu, and J. Tromp. Adjoint centroid-moment tensor inversions. *Geophys. J. Int.*, 186:264–278, 2011.